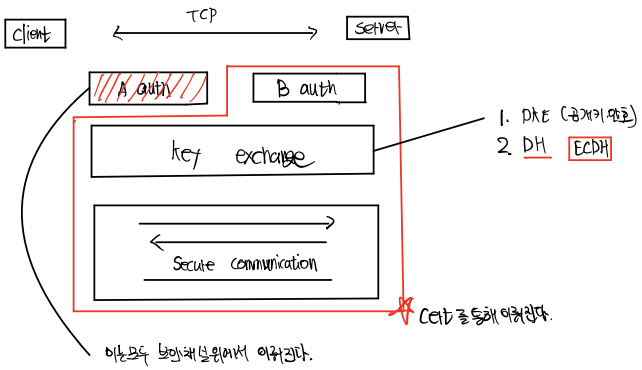


어느곳 보안채널위에서 이뤄지나.

ID/pw

OTP (보안카드)

cert (공인인증서)

Bio (생체인증)

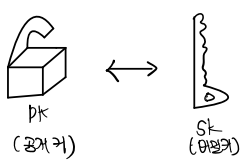
SNS/ARS/email

CHACHA (회원인증)

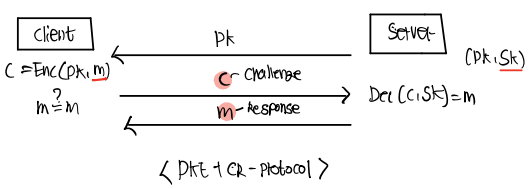
○ Server authentication

1. PKE (공개키암호)
2. Digital signature (전자서명)

+ challenge - Response protocol (프로토콜)



○ 공개키암호시스템

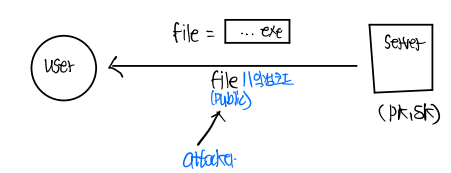


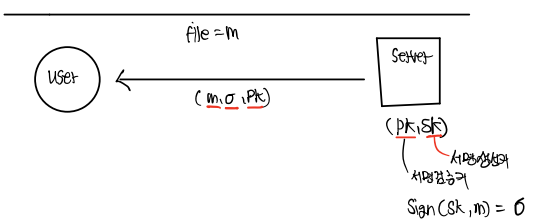
○ 전자서명

○ keyGen → (pk, sk)

○ Sign(sk, m) → σ (서명) (m, σ)

○ Verify(pk, m, σ) ⇒ 1/0 (valid / invalid)





Verify(pk, m, σ)

→ 1/0

1. ○ pk에 대응하는 사람이 m이라는 메시지에 대해 서명을 했다.
공격자가 메시지에 대한 서명X.
pk-소유자인 서버가 정상적서명을했음이 메시지를 보낸것이다. = 서버가 서명했다.

= server가 서명을 수행 (sender auth)

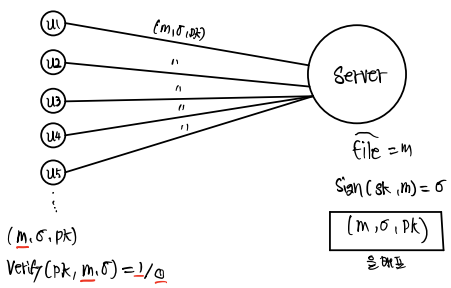
2. o 메시지를 받아서 검증을 했을때 정상적으로 검증
=> 메시지가 원본 그대로 서버에서 사용자에게 갔왔다.
= m이 원본 그대로 S->U 전송. (integrity = message auth)

(m, σ, pk)

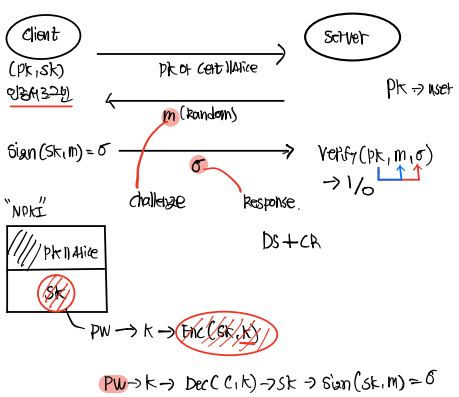
($\hat{m}$ = m인 상황에서, σ, pk) 가능할까? = X. (서명의 안전성)

1. RSA 서명
2. ECDSA 서명
   Elliptic Curve (EC)

1.

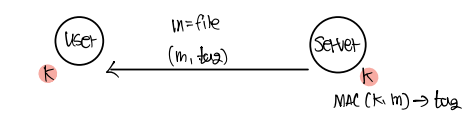


2. User auth



3. Server auth

· 대칭키 기반의 인증방법 MAC => {MAC, verify}

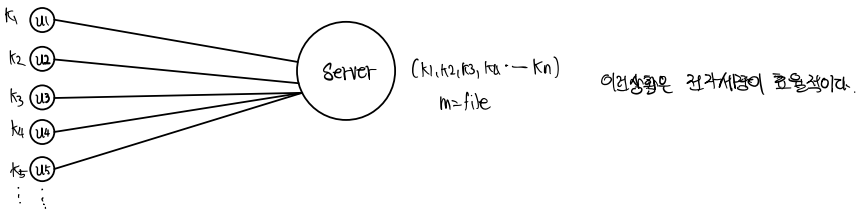


Verify(k, m, tag)
-> 1/0

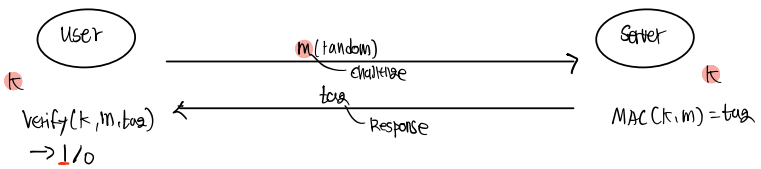
- MAC(k, m) -> tag
- verify(k, m, tag) -> 1/0

- k를 공유한 상대방이 tag를 만들었다. (Sender auth)
- m이 전송도중에 위조/변조 X (integrity = 메세지 무결성)

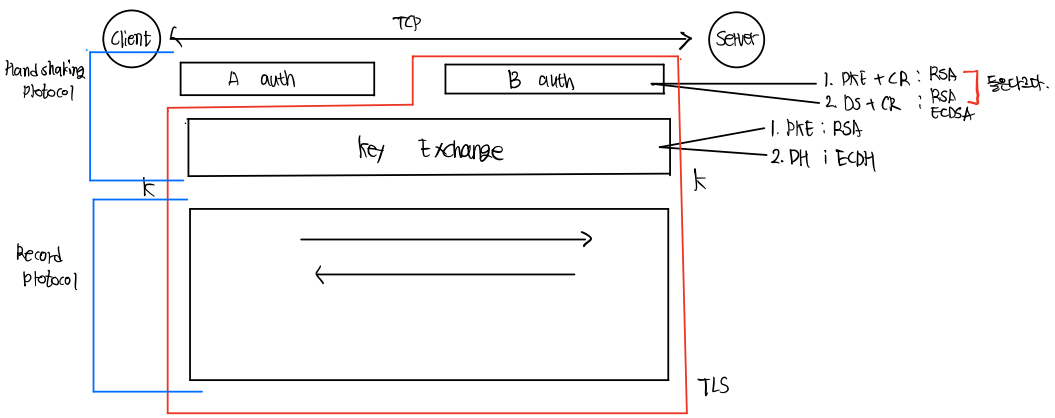
전자서명과의 차이는 k를 공유하고 있냐이다.



서버인증



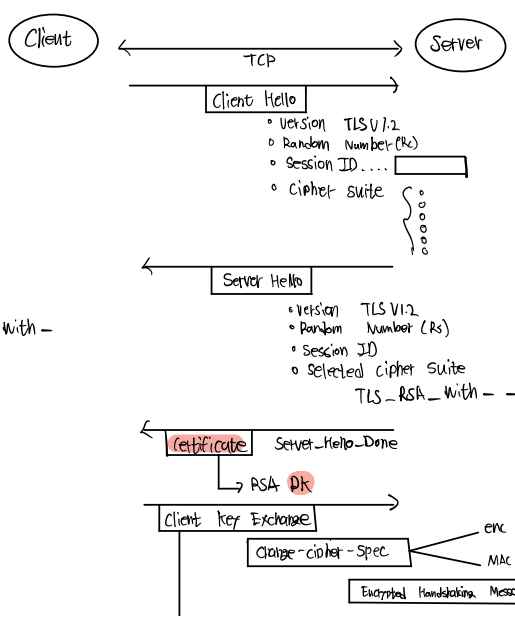
〈 MAC + CR - protocol 〉



- TLS _ RSA (DHE) _ with _ 대칭키암호 _ 해쉬함수
- TLS _ ECDHE _ {ECDSA / RSA} _ with _ 대칭키암호 _ 해쉬함수

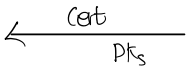
서버인증, 사용자인증, key exchange

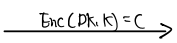
- TLS _ RSA _ with _ _



TLS _ RSA _ with _

서버인증





- $Enc(pk, k) = C$ (as a RSA encryption)
  - k: Premaster secret
    - → k_e, k_m

$MAC(k_m, R_c \| R_s) \to tag$ (MAC tag)

k - - - - - - - k

← Encrypted Handshake message ($\widetilde{tag}$) — k_m

$Verify(k_m, \widetilde{tag}, \sim)$

$\to 1/0$

Server는 $Dec(C, sk) \to k$ → k_e, k_m

$Verify(k_m, tag, R_c \| R_s)$ (사용자 인증 X)

$\to 1/0$

서버가 encrypted handshaking 이 제대로 만들어졌는지 검증.

← | Change-Cipher-Spec | Encrypted Handshaking message |

Change-Cipher-Spec: k_e, k_m을 쓰겠다 선언

Encrypted Handshaking message: $MAC(k_m, R_c \| R_s \| (////)) \to \widetilde{tag}$

(////): 이전 메시지의 hash값

Client는

k_e, k_m

$Verify(k_m, R_c \| R_s \| (////), \widetilde{tag})$

$\to 1/0$ → Server auth.

---

hand shake protocol Done

k_e, k_m ——— record Protocol ——— k_e, k_m

TLS_RSA_with_ _ 1.2

---

o TLS_ECDHE_{ECDSA / RSA}_with_ _

EC DHE

- EC: 타원곡선
- E: Ephemeral DH — 임시로 사용하는, 일회용

Client ←—— TCP ——→ Server

→ Client Hello

- Version TLS v1.2
- Random Number (R_c)
- Session ID ....
- Cipher suite { o o o o o o

← Server Hello

- Version TLS V1.2
- Random Number (R_s)
- Session ID
- Selected cipher suite
  - TLS_ECDHE_{ECDSA / RSA}_with_ _

← | Certificate | Server key Exchange | Server-Hello-Done |

Certificate → • pk (전자서명)

Server key Exchange → ECDH를 위한

$B = g^{y} \pmod{p} = m$ (DH의 일부)

σ

$Sign(sk, m) = \sigma$

(서버가 유저에게 패킷 전송시)

User ← (m, σ, pk) — Server

과정일하다.

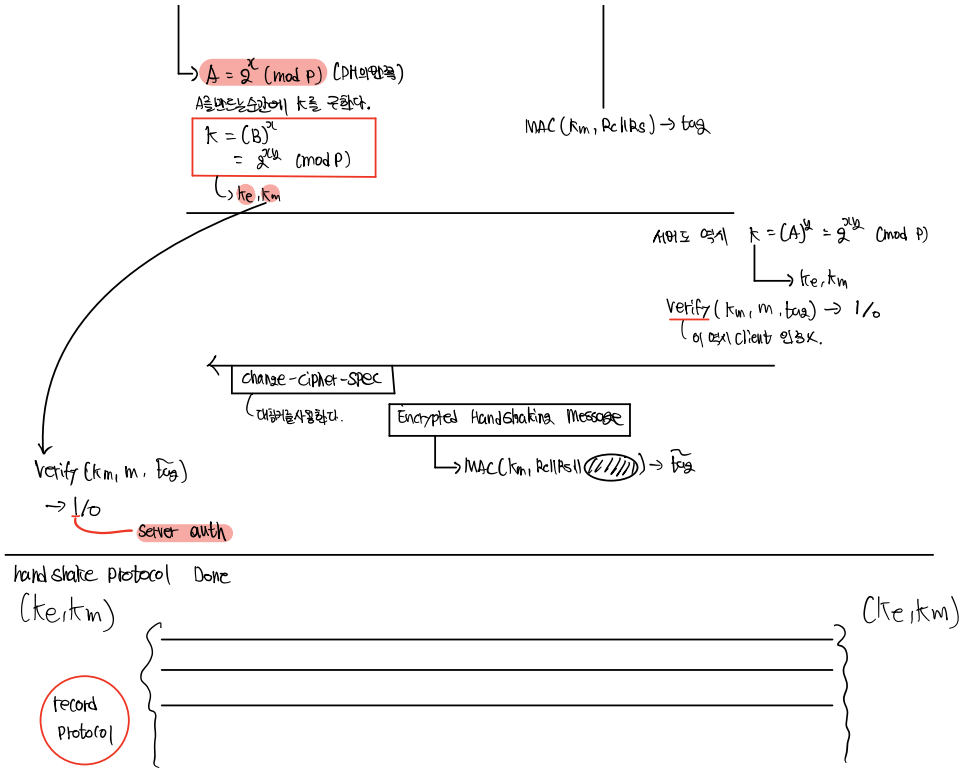
$Verify(pk, m, \sigma)$

$\to 1/0$

DS + DH + MAC

- DS: 인증은 불가능
- DH: 은닉키교환

→ | Client key Exchange | Chang-cipher-Spec | Encrypted Handshaking Message |



handshake Protocol Done